Finite Fields, Algebra, and Gabidulin Codes

Galileo Grey

July 2024

Presentation of the Laboratory & Personal Assessment of the Excellence Internship Program

The Institut Fourier is the pure mathematics laboratory of Université Grenoble Alpes, according to its website, its main research themes are:

- Algebra and Geometry
- Combinatorics
- Geometry and Topology
- Mathematical Physics
- Probability
- Number Theory.

Notably, during my time at the laboratory there was an event held on the mathematics of black holes, as well as the laboratory's yearly summer school, which this year was held on low-dimensional topology.

I and my fellow interns worked in the laboratory's library, which is very nice and has a wide selection of mathematical literature, including both journals and textbooks on specific topics (I had previously used the library's copies of "Winning Ways for your Mathematical Plays" and "On Numbers and Games" by John Conway for a presentation on combinatorial game theory).

Having already participated in the internship program last year at the Laboratoire Jean Kuntzmann of Applied Mathematics and Computing, I was definitely able to gauge how much I had progressed through my second year of my undegraduate degree. While last year I was only able to implement some of the ideas I found in the text my internship was based on ("Analyse Numérique et Équations Différentielles" by J.P. Demailly), this year I was able to understand the theory of the subject on a deeper level, while also applying it numerically.

The internship also helped me realise what appealed to me the most about pure math research, which were the most general results surrounding the subject. The real-world motivation was interesting and helped me explain what the internship was about to others, but it was the more theoretical results that I found most interesting.

Theoretical & Experimental Results

Numerical implementation of techniques described in this report are available at: https://github.com/GreyGalileo/IF-Internship

Self-correcting codes are encodings that attempt to ensure that information can be transmitted correctly despite possible interference during transmission which may change some of the information being sent.

Gabidulin codes in particular address cases such as antenna failure or momentary interference where the error produced affects the information in a predictable way (faulty information sent from a particular antenna/sent at a particular time).

To do so, coded messages are interpreted as a subset of vectors in a space $\mathbb{F}_{2^m}^n$, in such a way that the above mentioned errors will provoke only small changes in the rank of a matrix representation of the vector and the original message can thus be reconstructed by the receiver, provided the subset of valid codes and the rank of the incurred error are small enough.

1 The field $\mathbb{F}_{2^m}$

An explicit description of the field $\mathbb{F}_{2^m}$ is helpful for understanding these codes, since it will be necessary to build matrices and solve linear equations over $\mathbb{F}_{2^m}$ in order to decode messages sent using the Gabidulin encoding scheme.

1.1 Finding an irreducible polynomial

The field of size 2^m, $\mathbb{F}_{2^m}$, is isomorphic to quotient rings $\mathbb{F}_2[X]/(P)$ where P is an irreducible polynomial of degree m. Since arithmetic in this quotient ring depends on P, in order to properly represent $\mathbb{F}_{2^m}$, we must first find an irreducible polynomial in $\mathbb{F}_2[X]$ of degree m for any natural number $m \in \mathbb{N}_{\geq 2}$. Since any element in $\mathbb{F}_{2^m}$ that generates the multiplicative group $(\mathbb{F}_{2^m}^*, \times)$ is the root of some degree m irreducible polynomial, we know such a polynomial exists. To find one we will use Berlekamp's algorithm for finding the number of irreducible factors of a polynomial over a finite field.

1.1.1 Berlekamp's Algorithm

The complete version of Berlekamp's algorithm computes the irreducible factors of a polynomial P over a field $\mathbb{F}_q$, for our purposes we only need the first part of the algorithm, which finds the number of irreducible factors of P (P is irreducible when it has exactly 1 irreducible factor, itself). To find the number of irreducible factors of a polynomial P in $\mathbb{F}_q[X]$ the algorithm computes the matrix of the linear map [1] :

$$\begin{aligned} S_P : \mathbb{F}_q &\to \mathbb{F}_q \\ Q &\mapsto (Q^q - Q) \bmod P \end{aligned}$$

When P is irreducible, the ring $\mathbb{F}_q[X]/(P)$ is a field and the polynomial $Q^q - Q$ has at most q roots over the field

$$\mathbb{F}_q[X]/(P) \simeq \mathbb{F}_{q^{\deg(P)}}.$$

Its kernel is therefore exactly the subfield $\mathbb{F}_q$. Thus the dimension as an $\mathbb{F}_q$-space of the kernel is 1.

When P is square-free, the Chinese remainder theorem gives that the ring $\mathbb{F}_q[X]/(P)$ is isomorphic to the Cartesian product

$$\mathbb{F}_q[X]/(P_1) \times ... \times \mathbb{F}_q[X]/(P_n),$$

where $P_1...P_n$ are the irreducible factors of P.

In this case, since each ring in the Cartesian product is a field, the kernel of $Q^q - Q$ is 1-dimensional on each factor $\mathbb{F}_q[X]/(P_i)$, and the kernel of this map is the Cartesian product of n 1-dimensional spaces, so it is indeed n-dimensional. This argument is summarised in the following proposition:

Proposition 1 *For any square-free $P \in \mathbb{F}_q[X]$, the dimension of the kernel of S_P is the number of irreducible factors of P.*

If P is not square-free, it is obviously not irreducible. The dimension of the kernel of S_P can be calculated using Gaussian elimination.

1.1.2 Square-free polynomials

Berlekamp's algorithm requires a square-free polynomial. We will now describe how we can effectively check whether a given polynomial is square-free or not. To determine if a polynomial P is square-free, we can take the gcd of P and P', the derivative of P, which is defined as follows:

$$\Sigma_{i=0}^{n} a_i X^i \mapsto \Sigma_{i=0}^{n} i a_i X^{i-1}$$

We will begin by proving that the chain rule holds for such polynomials:

Proposition 2 *For any product of polynomials $P = QR$ with $P, Q, R \in \mathbb{F}_q[X]$, $P' = Q'R + QR'$.*

[1] The map $Q \mapsto Q^q$ is linear since for all $\omega \in \mathbb{F}_q$, and for all $Q, R \in \mathbb{F}_q[X]$,

$$(\omega Q + R)^q = \omega^q Q^q + R^q = \omega Q^q + R^q,$$

since all terms in the binomial expansion except these two have coefficients divisible by $q (\equiv 0 \bmod q)$, and any element of $\mathbb{F}_q$ is a fixed point of $\omega \mapsto \omega^q$. The map $Q \mapsto Q^q - Q$ is also linear, being a linear combination of the above map and the identity map.

Proof: Let $Q = \Sigma_{i=0}^{\deg(Q)} b_i X^i$ and $R = \Sigma_{i=0}^{\deg(R)} a_i X^i$, then we have that $P = \Sigma_{i=0}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} a_j b_{i-j}) X^i$. Then $Q' = \Sigma_{i=1}^{\deg(Q)} i a_i X^{i-1}, R' = \Sigma_{i=1}^{\deg(R)} i b_i X^{i-1}$ and $P' = \Sigma_{i=1}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} a_j b_{i-j}) i X^{i-1}$
Computing $Q'R + QR'$ gives:

$$\begin{aligned}
&Q'R + QR' \\
&= (\Sigma_{i=1}^{\deg(Q)} i a_i X^{i-1})(\Sigma_{i=0}^{\deg(R)} b_i X^i) + (\Sigma_{i=0}^{\deg}(Q) a_i X^i)(\Sigma_{i=1}^{\deg(R)} i b_i X^{i-1}) \\
&= (\Sigma_{i=1}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} j a_j b_{i-j}) X^{i-1} + (\Sigma_{i=1}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} i - j a_j b_{i-j}) X^{i-1} \\
&= (\Sigma_{i=1}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} (i - j + j) a_j b_{i-j}) X^{i-1} \\
&= (\Sigma_{i=1}^{\deg(Q)\,\deg(R)} (\Sigma_{j=0}^{i} a_j b_{i-j}) i X^{i-1} = P'
\end{aligned}$$

Lemma 1 *If P is square free, then the gcd P and P' is 1, otherwise, it will be the product of the irreducible factors of P, each with 1 less multiplicity than in P itself.*

Proof: For each irreducible factor P_i of $P = P_i^{\alpha_i} Q$, the chain rule gives that

$$P' = P_i^{\alpha_i} Q' + \alpha_1 P_i^{\alpha_i - 1} Q,$$

and as such P' is divisible by $P_i^{\alpha_i - 1}$ but not $P_i^{\alpha_i}$. Since all divisors of P can be written as products of these irreducibles, the gcd of P and P' is the product of the $P_i^{\alpha_i - 1}$. In particular, when all factors have multiplicity 1, this product is exactly 1.
Remark: The gcd of two polynomials can be computed using Euclid's algorithm. We will now discuss its complexity. Let m be the degree of the higher-degree polynomial. In the worst case, the gcd is 1 and each subtraction of polynomials reduces the degree of the result by 1, requiring $O(m)$ subtractions of polynomials which have degrees ranging from m to 1. The total time complexity of the gcd algorithm is $O(\frac{m^2}{2})$.

1.1.3 Complete algorithm

The full version of the modified Berlekamp's algorithm for determining if a polynomial is irreducible is shown in algorithm 1. We can successively pick polynomials and test them with this algorithm to find one that is irreducible.

1.1.4 Frequency of irreducible polynomials

Since our strategy is to test polynomials of degree m until we find one that is irreducible, we would like to know how many of these there are, as a function of m (i.e. how often we should expect to find an irreducible).

For any integer n, the field $\mathbb{F}_{2^n}$ consists of the roots of the polynomial $X^{2^n} - X$. Each element ω of $\mathbb{F}_{2^n}$ admits a minimal monic polynomial P_ω (all polynomials in $\mathbb{F}_2[X]$ are monic), such that any polynomial which admits ω as a

Algorithm 1 Determining whether a polynomial is irreducible

```
function IRREDUCIBLE($P$)
    $P' \leftarrow$ Derivative($P$)
    if gcd($P, P'$) $= 1$ then
        $num \leftarrow$ Berlekamp($P$) @returns number of irreducible factors
        if $num = 1$ then
            return True
        else
            return False
        end if
    else
        return False
    end if
end function
```

root also admits P_ω as a divisor, thus $\forall \omega \in \mathbb{F}_{2^n} P_\omega$ divides $X^{2^n} - X$. Additionally, we know that the degree of P_ω divides n, since $\mathbb{F}_2[X]/(P_\omega) \simeq \mathbb{F}_{2^{\deg((P_\omega))}}$ is a sub-field of $\mathbb{F}_{2^n}$, and field extensions are multiplicative with respect to degrees. Since $X^{2^n} - X$ splits over $\mathbb{F}_{2^n}$, all divisors of $X^{2^n} - X$ admit some $\omega \in \mathbb{F}_{2^n}$ as a root and so the irreducible divisors of $X^{2^n} - X$ are exactly the minimal polynomials of elements in $\mathbb{F}_{2^n}$, which all have degrees dividing n.

For an irreducible polynomial P of degree d where d divides n, we know that P is the minimal polynomial for some $a \in \mathbb{F}_{2^d} \simeq \mathbb{F}_2[X]/(P)$, and a, being a member of $\mathbb{F}_{2^d}$ is a fixed point of $x \mapsto x^{2^d}$ and therefore also of $x \mapsto x^{2^n}$, which makes it a root of $X^{2^n} - X$ and a member of $\mathbb{F}_{2^n}$.

We have that $X^{2^n} - X$ is divisible by all monic irreducible polynomials of degree d when d divides n, and furthermore that all monic irreducible divisors of $X^{2^n} - X$ have degree that divides n. $X^{2^n} - X$ is therefore exactly the product of all monic irreducible polynomials with degrees that divide n.

We write I_n for the number of monic irreducible polynomials of degree n in $\mathbb{F}_2[X]$. We then have that

$$2^n = \Sigma_{d|n} d I_d,$$

since each irreducible polynomial counted in I_d contributes d degree to $X^{2^n} - X$.

In order to find an explicit expression of I_n, we will use the Möbius inversion formula, stated below.

Theorem 1 *given two functions $f, g : \mathbb{N} \to \mathbb{C}$ such that $f(n) = \Sigma_{d|n} g(d)$,*

$$g(n) = \Sigma_{d|n} \mu(d) f(\frac{n}{d}) \tag{1}$$

where μ is the Möbius function defined as:

$$\mu(n) = \begin{cases} 1 & \textit{if } n \textit{ square-free with even number of prime factors*} \\ -1 & \textit{if } n \textit{ square-free with odd number of prime factors*} \\ 0 & \textit{if } n \textit{ admits a square factor} \end{cases}$$

** prime factors counted without multiplicity*

Proof: Expanding the terms $f(\frac{n}{d})$ sum on the right side of (1) gives:

$$\Sigma_{d|n}\mu(d)f(\frac{n}{d}) = \Sigma_{d|n}\mu(d)\Sigma_{k|\frac{n}{d}}g(k).$$

The integer n can be written as a product of primes $p_1^{\alpha_1}...p_j^{\alpha_j}$. Since every k that appears in the sum divides $\frac{n}{d}$ and n, k can also be written as $p_1^{\beta_1}...p_j^{\beta_j}$ with $\beta_i \leq \alpha_i$. The term $\mu(d)$ is only non-zero when d is square-free, that is to say it contains each $p_1, p_2, ...p_j$ at most once. The set of these divisors d for which μ is non-zero is isomorphic to *(bijective with?)* the powerset (set of subsets) of $\{p_1, p_2, ...p_j\}$. Counting the number of times $g(k)$ appears in the sum for a certain k is the same as counting how many divisors d of n contain only primes p_i such that $\alpha_i \neq \beta_i$. This, in turn, is the same as counting the subsets of the set $\{p_i | \alpha_i \neq \beta_i\}$, indeed, the products of the elements in such subsets are exactly the divisors d for which $g(k)$ appears in the sum $\Sigma_{k|\frac{n}{d}}g(k)$.

When a finite set S is non-empty, it has the same number of subsets containing an even and odd number of elements. This can be justified using any map that, for a certain member $x \in S$, sends subsets Y not containing x to $Y \cup \{x\}$ and sends subsets Z containg x to $Z \setminus \{x\}$. Any such map is a bijection between subsets of S containing an even number of elements and those which contain an odd number of elements, so there are the same number of each. Since $\mu(d)$ is 1 when d has an even number of prime factors and -1 when it has an odd number, and $g(k)$ appears the same amount of times in both cases, the terms $g(k)$ cancel for all $k \neq n$. The term $g(n)$, on the other hand, only appears once, for the divisor 1 isomorphic *(/bijective/mapped?)* in this model to the empty set of prime factors), and so appears positively in the sum. Since all terms except $g(n)$ are cancelled,

$$\Sigma_{d|n}\mu(d)f(\frac{n}{d}) = \Sigma_{d|n}\mu(d)\Sigma_{k|\frac{n}{d}}g(k) = g(n),$$

and the Möbius inversion formula holds.

The Möbius inversion formula implies that

$$nI_n = \Sigma_{d|n}\mu(d)2^{\frac{n}{d}}$$

. This value can be bounded above by $2^n + 2^{\log_2(n)}2^{\frac{n}{2}}$ and below by $2^n - 2^{\log_2(n)}2^{\frac{n}{2}}$ since all primes are ≥ 2 and n has fewer than $\log_2(n)$ prime factors. Since $2^n = o(n2^{\frac{n}{2}})$, by the sandwich theorem $I_n \sim \frac{2^n}{n}$.

1.2 Arithmetic over $\mathbb{F}_{2^m}$

Before implementing algorithms in $\mathbb{F}_{2^m}$, we must first implement the elementary operations over this field using the irreducible polynomial P we have found (such that $\mathbb{F}_{2^m} \simeq \mathbb{F}_2[X]/(P)$).

We represent members of $\mathbb{F}_{2^m}$ as vectors in $\mathbb{F}_2^m$, or as the coefficients of polynomials in $\mathbb{F}_2[X]$ with degree $< m$. Numerically this means length m arrays of 1s and 0s.

Comment: Since the elements being represented are binary stings, it would be much more efficient to represent them using the bit representation of 64-bit integers, such that, for instance, addition can be done with 1 XOR instruction. Using this method will not, however, change the time complexity of our algorithms, in fact, for large m, it is useful to think of the elements as being arrays, such that we can assign values at certain indices without affecting the rest of the bits.

1.2.1 Addition

Using the above representation, addition is simple. Since the field is of characteristic 2, adding vectors or polynomials consists only of taking the exclusive or of each pair of entries. Being of characteristic 2, every element in $\mathbb{F}_{2^m}$ is its own additive inverse.

1.2.2 Multiplication

Multiplication of elements in $\mathbb{F}_{2^m}$ is done by repeated shifting and addition and then taking the remainder modulo P:

Algorithm 2 Multiplication in $\mathbb{F}_{2^m}$

```
function MULTIPLY($\omega_1, \omega_2$)
    $s \leftarrow 0$
    $h \leftarrow \omega_1$
    for $c$ in $\omega_2$ do @for loop starts at coefficient with lowest degree
        if $c = 1$ then $s \leftarrow s + h$
        end if
        $h \leftarrow h << 1$ @left shift adds 0 to the end of the array, analogous to multiplication by 2
    end for
    return remainder_euclidean_division($s, P$)
end function
```

1.2.3 Squaring

While squaring an element of $\mathbb{F}_{2^m}$ could be done by multiplying it by itself, there is a much more efficient method. Since over $\mathbb{F}_2$, we have

$$(\Sigma_{i=0}^{m} a_i X^i)^2 = \Sigma_{i=0}^{m} a_i X^{2i}$$

, instead of using the shift and addition strategy we used with multiplication, we can simply add 0s in between each coefficient in the list (thus multiplying the degree of the term by 2 in a list of size $2m$) and then taking the remainder modulo P as before.

1.2.4 Multiplicative Inverses

Taking the inverse over $\mathbb{F}_{2^m}$ is not as obvious, a possible strategy is to say that every member of $\mathbb{F}_{2^m}$ is a root of $X^{2^m} - X$ so for all $\omega \in \mathbb{F}_{2^m}$, there exists $a \in [1, m]$ such that $\omega^{2^a} = \omega$.

For such an a, we know that $\omega^{2^a-2} = \omega^{-1}$, we can iteratively compute both ω^{2^k} and ω^{2^k-2} using the identities $(\omega^{2^k})^2 = \omega^{2^{k+1}}$ and $\omega^{2^k-2}\omega^{2^k} = \omega^{2^{k+1}-2}$ and then return the latter value when the former reaches ω.

Algorithm 3 Inversion in $\mathbb{F}_{2^m}$

```
function INVERSE($\omega$)
    $\omega^{2^k} \leftarrow$ square($\omega$) @squaring over $\mathbb{F}_{2^m}$
    $\omega^{2^k-2} \leftarrow 1$
    while $\omega^{2^k} \neq \omega$ do
        $\omega^{2^k-2} \leftarrow \omega^{2^k} \times \omega^{2^k-2}$ @multiplication over $\mathbb{F}_{2^m}$
        $\omega^{2^k} \leftarrow$ square($\omega^{2^k}$)
    end while
    return $\omega^{2^k-2}$
end function
```

In the worst case (when $\text{ord}(\omega) = 2^m - 1$), this algorithm takes $m - 1$ multiplications and m squarings.

2 Gabidulin Codes

We now want to consider the Gabidulin codes themselves, although the messages being sent are sequences of 1s and 0s, these are interpreted as vectors in the space $\mathbb{F}_{2^m}^n$, which can also be seen as matrices over the field $\mathbb{F}_2$

2.1 The rank metric

In order to be able to decode an erroneous message, we must be able to find the 'closest' valid Gabidulin code, to do this, we should define a metric on the space $\mathbb{F}_{2^m}^n$, which should give some notion of how many errors of the kind we expect (errors along rows and columns) it would take to go from some valid Gabidulin code to the distorted message. The following lemma relates the two representations of codes mentioned above.

Lemma 2 *The rank of $c = (c_1, ..., c_n) \in \mathbb{F}_{2^m}^n$ is the same as the dimension of the $\mathbb{F}_2$-vector subspace of $\mathbb{F}_{2^m}$ generated by the vectors $c_1, ..., c_n$.*

Proof: We take a basis e' of $\mathrm{Vect}(c_1, ..., c_n)$ which can be completed to a basis e of $\mathbb{F}_{2^m}$ and write $r = \dim(\mathrm{Vect}(c_1, ..., c_n))$. Writing the matrix of c in the basis e gives all 0 entries after the first r lines since the vectors in c are generated by the first r members of the basis $e' \subset e$. Since the vectors in c generate the same subspace as e', the rank of the matrix is exactly r and the rank is the same as the dimension of the subspace.

Definition 1 *The distance bewteen a message and a valid Gabidulin code is equal to the rank of their difference, interpreted as matrices over* $\mathbb{F}_2$*.*

We will prove that this definition provides a deistance over the sapce in trhe classical sense.

1. The distance between two vectors is 0 iff they are the same vector:

$$\forall x, x' \in \mathbb{F}_{2^m}^n, d_r(x, x) = \mathrm{rank}(0) = 0;$$
$$d_r(x, x') = 0 \implies \mathrm{rank}(x - x') = 0 \implies x = x'$$

2. The function is symmetric since multiplication by -1 does not change the rank (and anyways we are working with fields of characteristic 2:

$$\forall x, y \in \mathbb{F}_{2^m}^n, d_r(x, y) = \mathrm{rank}(x - y) = \mathrm{rank}(y - x) = d_r(y, x)$$

3. The triangular inequality holds:
$\forall x, y, z \in \mathbb{F}_{2^m}^n, \mathrm{Vect}(x - z) \subset \mathrm{Vect}(x - y) + \mathrm{Vect}(y - z)$, since any vector in the former space can be constructed by a combination of vectors in the other two. Thus, by Lemma 2 we have that

$$d_r(x-y)+d_r(y-z) = \mathrm{rank}(x-y)+\mathrm{rank}(y-z) \leq \mathrm{rank}(x-z) = d_r(x-z)$$

Definition 2 *The minimal distance of a subset $\mathcal{C}$ of a vector space V with respect to a distance d is* $d_{min} := \min\{d(x, y)|x, y \in \mathcal{C}, x \neq y\}$

The minimal distance between two valid codes will be important for determining what kind of errors are or aren't correctable using Gabidulin codes.

2.2 Structure of $\mathbb{F}_2$-linear polynomials

To understand which vectors are valid Gabidulin codes, it is useful to first study a certain subset of polynomials in $\mathbb{F}_2[X]$, namely those that are also linear maps over the vector space $\mathbb{F}_{2^m}[X]$.

Definition 3 *An $\mathbb{F}_2$-linear polynomial is a polynomial which has non-zero coefficients on terms with degrees which are powers of two, they can be written as $P(X) = \Sigma_{i=0}^{n} a_i X^{2^i}$ where $n \in \mathbb{N}$ and $a_i \in \mathbb{F}_{2^m}$.*

Since each term in the finite sum can be thought of as a repeated squaring of X, and we have shown above that squaring is linear in fields of characteristic two, it is easy to show that $\mathbb{F}_2$-linear polynomials are also linear maps.

Let P be a $\mathbb{F}_2$-linear polynomial over $\mathbb{F}_{2^m}$ of degree 2^k. P is a linear endomorphism so its roots form a vector subspace of $\mathbb{F}_{2^m}$. The dimension of $\mathrm{Ker}(P)$ is at most k over $\mathbb{F}_2$ because it has at most 2^k roots and that is precisely the cardinality of a k-dimensional $\mathbb{F}_2$-vector space, therefore:

Lemma 3 *The dimension of the kernel of a $\mathbb{F}_2$-linear polynomial of degree 2^k is at most k.*

Proposition 3 *The set of $\mathbb{F}_2$ -linear polynomials over $\mathbb{F}_{2^m}$ equipped with addition and function composition is a non-commutative ring.*

Proof: The fact that $\mathbb{F}_2$ linear polynomials form a vector space over $\mathbb{F}_{2^m}$ gives us that that they form an Abelian group with respect to addition.

We have only to prove now that function composition satisfies the structure of a ring: We can already say that function composition is associative, and that the polynomial $X \in \mathbb{F}_{2^m}[X]$ is the identity for composition. Composition is distributive over addition:

$$\begin{aligned}\forall P, Q, R \in \mathbb{F}_{2^m}[X], P \circ (Q + R) &= \Sigma_{i=0}^{k} a_k (Q + R)^{2^k} \\ &= \Sigma_{i=0}^{k} a_k Q^{2^k} + a_k R^{2^k} = P \circ Q + P \circ R\end{aligned}$$

for some sequence (a_k) taking values in $\mathbb{F}_{2^m}$. As before, since the exponent is a power of 2, all terms in the binomial expansion save the first and last are multiples of even scalars, which in a field of characteristic 2 makes them 0. The space is closed by composition since raising a polynomial to a power 2^k over this field only has the effect of multiplying the degree of all monomials by 2^k, maintaining $\mathbb{F}_2$-linearity.

Proposition 4 *The ring of $\mathbb{F}_2$-linear polynomials is a commutative ring if and only if $\mathbb{F}_{2^m} = \mathbb{F}_2$.*

($\Longrightarrow$) Consider the ring of $\mathbb{F}_2$-linear polynomials over $\mathbb{F}_2$, the coefficients of theses polynomials can only be 0 or 1, thus composition of polynomials in this ring takes the form of

$$\begin{aligned}(\Sigma_{i=0}^{k} X^{2^i}) \circ (\Sigma_{j=0}^{k} X^{2^j}) &= \Sigma_{i=0}^{k} (\Sigma_{j=0}^{k} X^{2^j})^{2^i} = \Sigma_{i=0}^{k} \Sigma_{j=0}^{k} X^{2^j 2^i} \\ &= \Sigma_{j=0}^{k} (\Sigma_{i=0}^{k} X^{2^i})^{2^j} = (\Sigma_{j=0}^{k} X^{2^j}) \circ (\Sigma_{i=0}^{k} X^{2^i})\end{aligned}$$

Note: i and j may skip some values (i.e. there may be terms with coefficient 0, this does not change commutativity.
($\Longleftarrow$) Assume that for some $\mathbb{F}_{2^m}$ the ring in question is commutative. Let

$\omega \in \mathbb{F}_{2^m}$, we have that

$$\begin{aligned} X^2 \circ \omega X &= \omega X \circ X^2 \\ \implies \omega X^2 &= \omega^2 X^2 \\ \implies \omega &= \omega^2 \\ \implies \omega &\in \{0,1\} \end{aligned}$$

Thus $\mathbb{F}_{2^m} = \mathbb{F}_2$

2.3 Decoding and Error Correction

Let $\mathbf{g} = (g_1, ..., g_n)$ be a vector over $\mathbb{F}_{2^m}$, with $(g_1, ..., g_n)$ linearily independent over $\mathbb{F}_2$ and let $k \in \mathbb{N}$,

$$Gab_k(\mathbf{g}) = \{(P(g_1), ..., P(g_n)) | P \text{ is } \mathbb{F}_2\text{-linear with degree } < 2^k\}$$

is the set of valid messages encoded with the Gabidulin encoding of order k around the vector $\mathbf{g}$.

Note: The constants m, n determine the size of the message being sent, which contains exactly $m \times n$ bits. The constant k determines the size of the subset of valid messages, as k grows, so do the number of valid messages and therefore the amount of information that can be sent. We will see later that having large k also makes it impossible to correct errors with larger rank.

When decoding, both the sender and reciever of the messages are assumed to know the vector g and the order k of the encoding, if either of these are not known, decoding is impossible.

Lemma 4 *The minimal distance of $Gab_k(\boldsymbol{g})$ with respect to the rank metric is $n - k + 1$.*

Proof: For two $\mathbb{F}_2$-linear polynomials $P_1 \neq P_2$ of degree $< 2^k$ the difference $P_1 - P_2$ has at most 2^{k-1} roots (assuming $P_1 \neq P_2$), so the kernel of the linear map $P_1 - P_2$ has dimension at most $k - 1$ over F_2 and so has rank at least $n - k + 1$. Since $\mathbf{g} = (g_1, ..., g_n)$ is composed of linearly independent vectors the difference $(P_1(g_1) - P_2(g_1), ..., P_1(g_1) - P_2(g_1))$ also has rank at least $n - k + 1$. Therefore this is the minimal

Take some initial message c and some recieved message $y = c + e$ with $e = (e_1, ..., e_n)$ of $\mathbb{F}_2$-rank t (recall that e can be seen as a matrix over an $\mathbb{F}_2$ vector space), where $c = (P(g_1), ..., P(g_n))$ for some $\mathbb{F}_2$-linear polynomial P, we want to prove the existence of a $\mathbb{F}_2$-linear polynomial V of degree at most 2^t such that $V(y_i) = V \circ P(g_i)$.

Since e has rank t, there is a basis of the subspace generated by e containing t vectors. Applying the following algorithm to this basis gives us a polynomial whose kernel is the space generated by e:

For a family of linearly independent vectors $(b_1, ..., b_k)$

$$V_1(x) = x^2 - b_1$$

for all $i \in [2, k]$, $$V_i(x) = (x^2 - P_{i-1}(b_i)x) \circ P_{i-1}(x)$$

V_k has degree 2^k and has kernel $\mathrm{Vect}(b_1, ..., b_k)$

For such a vector space endomorphism we have

$$V(y_i) = V(c_i) + V(e_i) = V(P(g_i)) = V \circ P(g_i)$$

To find the original message, we need to finding some pair $(V, N) \in \mathbb{F}_{2^m}$, with

$$\forall i \in \{1, .., n\}; \quad V(y_i) = N(g_i),$$

which is equivalent to solving a linear system of equations over $\mathbb{F}_{2^m}$ with n equations and $2t + k + 1$ unknown values in $\mathbb{F}_{2^m}$.

2.3.1 Upper bound on the error

Obviously if the interference has sufficiently large rank (for example if it transforms one valid message into a different valid message) it will be impossible to retrieve the original, we would therefore like to find an upper bound on the rank of the error introduced, below which the message containing the error will still be recoverable.

If the rank of the error is $t > (n - k)/2$ then the received word is more than half of the minimal distance $n - k + 1$ away from the solution, in this case there could be multiple words in $\mathrm{Gab}_k(g)$ that are sufficiently close to the received word to allow a solution, there is no reason to pick one solution over another and no way to retrieve the original message.